

# Cerebral LSTM: A Better Alternative for Single- and Multi-Stacked LSTM Cell-Based RNNs

Ravin Kumar\*

10 March 2020

## Abstract

Deep learning has rapidly transformed the natural language processing domain with its recurrent neural networks. LSTM is one such popular repeating cell unit used for building these recurrent neural network-based deep learning architectures. In this paper, we proposed a significantly improved version of LSTM named Cerebral LSTM which has much better ability to understand time-series data. Extensive experiments were conducted to get an unbiased performance comparison of our proposed version. Obtained results showed that recurrent neural network constructed using single Cerebral LSTM cell out-performed both recurrent neural network with single LSTM cell and recurrent neural network with two-stacked LSTM cells. We have also provided the PyTorch implementation of Cerebral LSTM [1].

**Keywords**— Recurrent Neural Network, Long Short-Term Memory, LSTM, Stacked LSTM, Natural Language Processing

## 1 Introduction

Long short-term memory [2] has accelerated the research work for problems based on time-series data by providing solution to vanishing and exploding gradient problems of recurrent neural networks [3]. LSTM is a special type of block which requires cell state  $C(t-1)$  and hidden state  $h(t-1)$  along with input data  $i(t)$  at each timestamp  $t$  to perform its operations. Fundamentally, LSTM consists of three type of gates, namely forget gate  $f(t)$ , input gate  $i(t)$  and output gate  $o(t)$  which decides relevant and irrelevant information from the input data (Fig. 1).

$$\begin{aligned}f(t) &= \sigma(W_f \cdot [h(t-1), x(t)] + b_f) \\i(t) &= \sigma(W_i \cdot [h(t-1), x(t)] + b_i) \\C_{tmp}(t) &= \tanh(W_c \cdot [h(t-1), x(t)] + b_c) \\C(t) &= f(t) * C(t-1) + i(t) * C_{tmp}(t)\end{aligned}$$

---

\*Affiliation: Department of Computer Science, Meerut Institute of Engineering and Technology, meerut-250005, Uttar Pradesh, India  
Email: [ravin.kumar.cs.2013@miet.ac.in](mailto:ravin.kumar.cs.2013@miet.ac.in), ORCID: 000-0002-3416-2679

$$o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o)$$

$$h(t) = o(t) * \tanh(C(t))$$

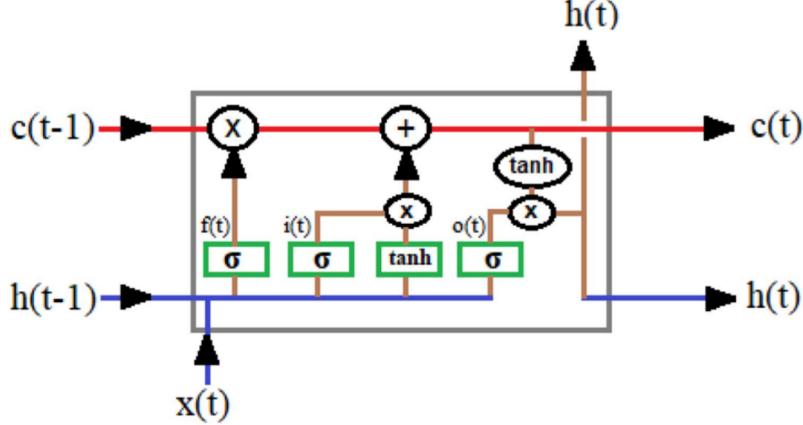


Figure 1: Architecture of long-short-term memory (LSTM) cell.

Forget gate decides which previous information  $C(t-1)$  is not required at the moment, input gate selects relevant information from the input data  $x(t)$ , and output gate produces the new hidden state  $h(t)$  for time  $t$ . At each timestamp  $t$ ,  $h(t)$  also serves as the output produced by the long short-term cell for timestamp  $t$ .

In this paper, we have proposed a new type of recurrent cell ‘‘Cerebral LSTM’’. To show effectiveness of our proposed cell, we have conducted experiments to perform its comparative analysis with LSTM-based recurrent neural networks. We have also shared the pytorch [4] implementation [1] of Cerebral LSTM in github.

## 2 Related Works

Hochreiter et al. [2] proposed a solution for understanding long-term dependencies in recurrent neural network. Chung et al. [5] designed a recurrent unit named GRU, having performance similar to LSTM. Bidirectional LSTM developed by Graves et al. [6] showed better performance in understanding time-series data than unidirectional LSTM. Cheng et al. [7] utilized a mechanism proposed by Srivastava et al. [8] in his research work to optimize the performance of LSTM. LSTM-based recurrent neural networks were used in designing many end-to-end deep learning solutions. Huang et al. [9] used two layers LSTM based generative model for music generation. In speech recognition, Graves et al. [10] used LSTM based recurrent neural network for better performance on TIMIT phoneme recognition. Sutskever et al. [11] used LSTM cells as a basic unit in recurrent neural network of both encoder and decoder parts of sequence-to-sequence model for performing language translation. Even in other cross-domain task such as image captioning [12], LSTMs are used in decoder part for generating textual description of the input image.

### 3 Recurrent Neural Networks

In the field of deep learning, neural networks had helped in solving many problems, but they were unable to analyze the time-series data. This problem leads to the development of a new type of neural network family called ‘recurrent neural networks’ (Fig. 2). With further research in the field, LSTM and then later GRU-based recurrent cells were introduced to solve the vanishing and exploding gradient problem of a simple recurrent neural network (Fig. 3).

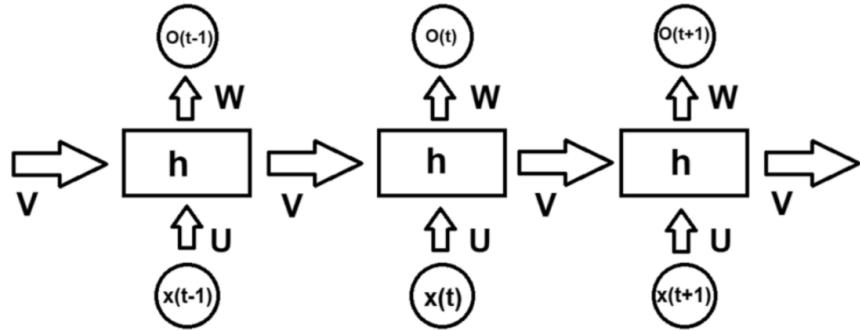


Figure 2: Architecture of recurrent neural networks.

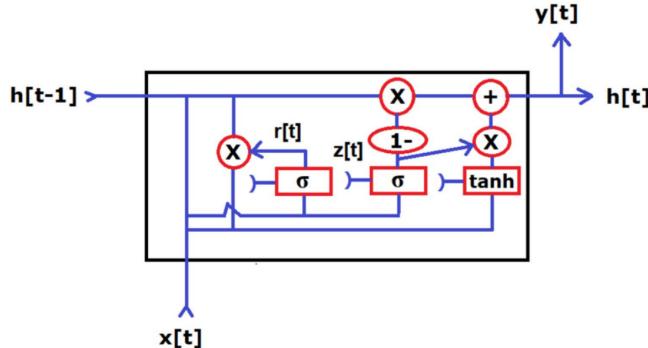


Figure 3: Architecture of a gated recurrent unit (GRU) cell.

Many end-to-end deep learning architectures were developed using recurrent neural networks to efficiently solve the problems related to time-series data. For better analysis of large time-series data, mechanisms of stacking (Fig. 4), and bidirectional RNN cells (Fig. 5) were developed.

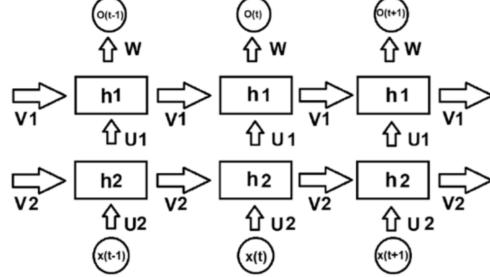


Figure 4: Architecture of two-stacked RNN cells.

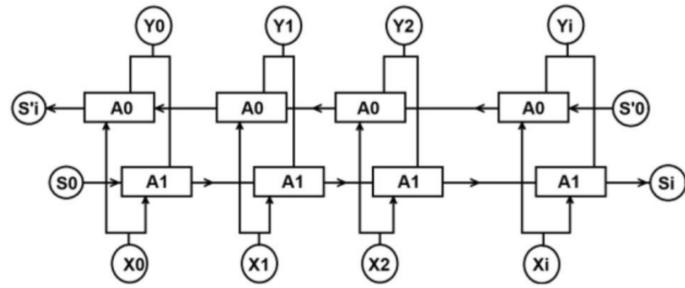


Figure 5: Architecture of bidirectional RNNs.

Specially, the development of sequence-to-sequence model [11] provided a huge boost in the field of natural language processing by providing end-to-end deep learning solution of various problem statements including language translation and designing conversational agents (Fig. 6).

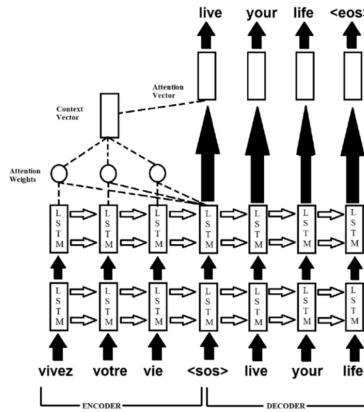


Figure 6: Attention-based sequence-to-sequence model for language translation.

## 4 Cerebral LSTM

Our proposed recurrent unit cell consists of one hidden state  $h(t)$  and two cell states  $U_c(t)$  and  $L_c(t)$ , where for each timestamp  $t$  we provide input  $x(t)$  with hidden state  $h(t-1)$  and cell states  $U_c(t-1)$  and  $L_c(t-1)$ .

Instead of sequentially processing layers, the Cerebral LSTM processes two parallel paths, reducing the risk of vanishing gradients and improving gradient flow.

It is called ‘‘Cerebral LSTM’’ because of the similarities present in abstract architecture of cerebral hemispheres of human brain and our proposed cell (Fig. 7, Table 1). One can also see that we have named functions starting with  $U$  to represent upper part, and  $L$  to represent lower part of Cerebral LSTM cell.

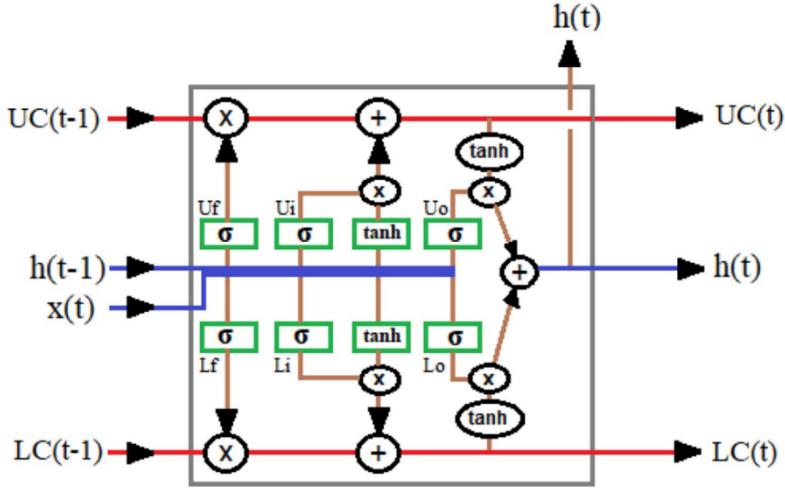


Figure 7: Architecture of our proposed Cerebral LSTM cell.

S. No.	function	Description
1	$U_c(t)$	Upper cell state at timestamp ‘t’
2	$L_c(t)$	Lower cell state at timestamp ‘t’
3	$U_f(t)$	Upper forget gate at timestamp ‘t’
4	$L_f(t)$	Lower forget gate at timestamp ‘t’
5	$U_i(t)$	Upper input gate at timestamp ‘t’
6	$L_i(t)$	Lower input gate at timestamp ‘t’
7	$U_o(t)$	Upper output gate at timestamp ‘t’
8	$L_o(t)$	Lower output gate at timestamp ‘t’
9	$h(t)$	Hidden state at timestamp ‘t’
10	$x(t)$	Input data at timestamp ‘t’

Table 1: Cerebral LSTM: Function-related details.

Mathematical Representation of Cerebral LSTM cell is given below:

$$\begin{aligned}
U_f(t) &= \sigma(W_{uf} \cdot [h(t-1), x(t)] + b_{uf}) \\
U_i(t) &= \sigma(W_{ui} \cdot [h(t-1), x(t)] + b_{ui}) \\
U_{Ctmp}(t) &= \tanh(W_{uc} \cdot [h(t-1), x(t)] + b_{uc}) \\
U_c(t) &= U_f(t) * U_{Ctmp}(t) + U_i(t) * U_{Ctmp}(t) \\
U_o(t) &= \sigma(W_{uo} \cdot [h(t-1), x(t)] + b_{uo}) \\
L_f(t) &= \sigma(W_{lf} \cdot [h(t-1), x(t)] + b_{lf}) \\
L_i(t) &= \sigma(W_{li} \cdot [h(t-1), x(t)] + b_{li}) \\
L_{Ctmp}(t) &= \tanh(W_{lc} \cdot [h(t-1), x(t)] + b_{lc}) \\
L_c(t) &= L_f(t) * L_{Ctmp}(t) + L_i(t) * L_{Ctmp}(t) \\
L_o(t) &= \sigma(W_{lo} \cdot [h(t-1), x(t)] + b_{lo}) \\
h(t) &= U_o(t) * \tanh(U_c(t)) + L_o(t) * \tanh(L_c(t))
\end{aligned}$$

Two separate memory cell states (i.e.  $U_c(t)$  and  $L_c(t)$ ) allows Cerebral LSTM for capturing more complex and diverse patterns in the data. The final hidden state  $h(t)$  leverages information from both  $U_c(t)$  and  $L_c(t)$  cell states, which allows model to capture more diverse features in the data.

In human brain, longitudinal fissure separates the cerebral hemispheres into left and right cerebral hemispheres (Fig. 8). Similarly, Cerebral LSTM consists of two cell states:  $U_c$  and  $L_c$  connected to same input  $x(t)$  and hidden state  $h(t-1)$  to update their cell states ( $U_c(t)$  and  $L_c(t)$ ) and jointly determine the updated value of the hidden state  $h(t)$ .

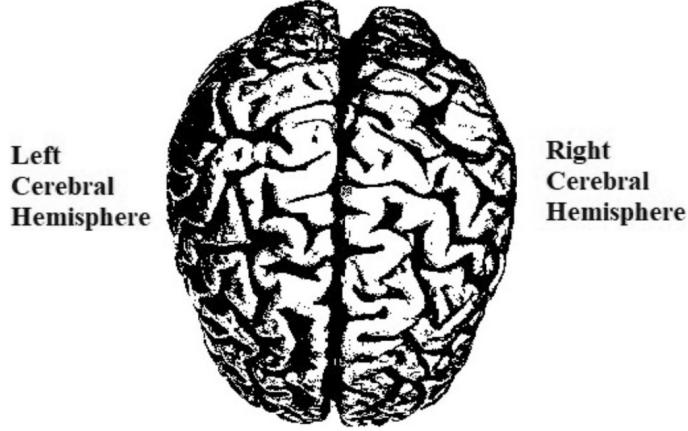


Figure 8: Cerebral hemispheres of human brain.

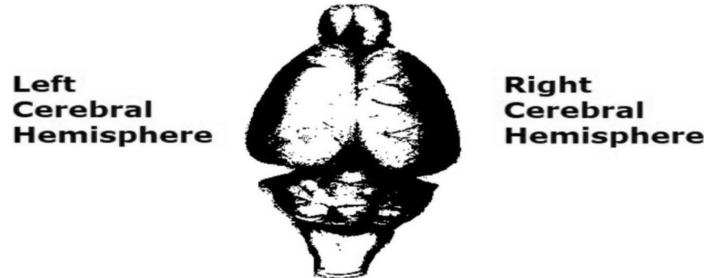


Figure 9: Cerebral hemispheres of a rat brain.



Figure 10: Cerebral hemispheres of a sheep brain.



Figure 11: Cerebral hemispheres of a chimpanzee brain.

#### 4.1 Impact of Initialisation of Trainable Parameters in Cerebral LSTM

The initial value of trainable parameters of upper and lower parts have impact on number of epochs required to train Cerebral LSTM cell. Ideally, upper and lower parts should not have same initial values for their trainable parameters.

#### 4.1.1 Identical initial trainable parameter values for upper and lower parts

Initial Symmetry: Upper and lower parts of the Cerebral LSTM process inputs identically, leading to similar cell states  $U_c(t)$  and  $L_c(t)$ .

Redundancy: Initial representations of upper and lower parts are redundant, potentially under-utilizing the model's capacity.

Gradients: Early training updates are similar, but divergence may occur over time, leading to different feature extraction.

#### 4.1.2 Different initial trainable parameter values for upper and lower parts

Diverse Learning: Upper and lower parts of Cerebral LSTM immediately capture different aspects of the data, enhancing representation diversity.

Specialization: Faster convergence and better utilization of the dual-path architecture, as each path can specialize in different features.

Performance: Improved performance due to richer, non-redundant representations from the start.

## 5 Comparative Analysis

We performed a comparative study on the performance of single-LSTM and two-stacked LSTM with respect to the performance of our proposed cell using Simpson dataset [13] and then analyzed the quality of data generated by each model. To obtain unbiased results, some parameters were made constant in each comparison (Table 2).

S. No.	Variable Name	Default Value
1	Learning rate	0.05
2	Epoch	500
3	Batch size	128
4	Weight initializer	Xavier initializer
5	Optimization	Adagrad optimizer
6	State size	512

Table 2: Default Values of Variables.

### 5.1 Comparative Study of single LSTM with two-Stacked LSTM on the Dataset

We first studied the behavior of recurrent neural networks based on single LSTM cell and on two-stacked LSTM cells and then made comparison on the basis training loss (Fig. 12).

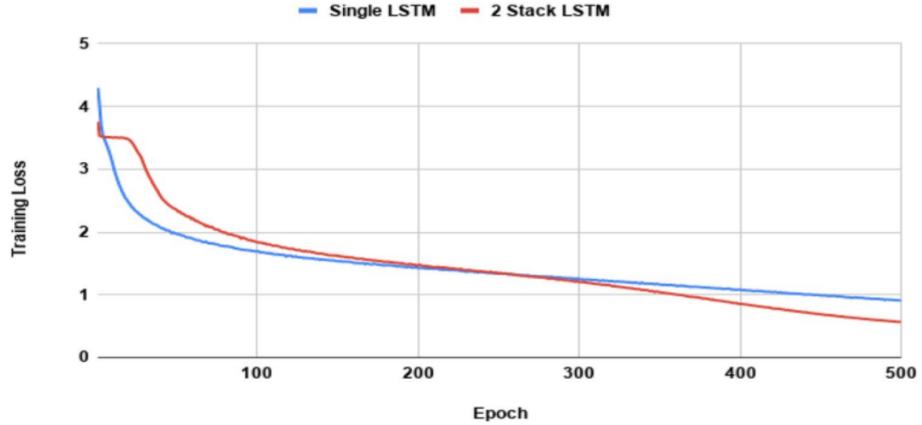


Figure 12: Comparative study of training loss.

After 250 epochs, two-stacked LSTM based recurrent neural network started performing better than single LSTM based recurrent neural network. This assures that dataset which we are using for comparative analysis is of good quality to perform further comparisons because common notion is that two-stacked LSTM should outperform single LSTM based recurrent neural network on dataset of considerable size.

## 5.2 Comparative Study of single LSTM with Cerebral LSTM

Our proposed Cerebral LSTM showed lower training loss from the beginning as compared to recurrent neural network based on single LSTM cell which helps it better understand time-series dataset.

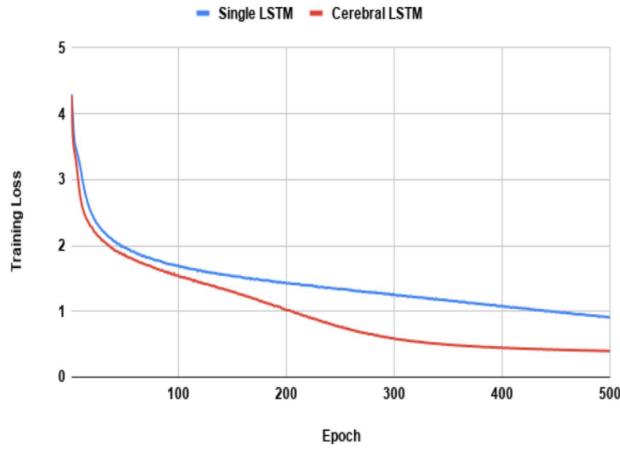


Figure 13: Comparative study of training loss.

Up to 250 epochs, traditional LSTM showed lower training loss than two-stacked LSTM in the previous comparative study. When single LSTM cell is compared with our proposed Cerebral LSTM, it is showed that Cerebral LSTM completely outperforms single-LSTM-based recurrent neural network and maintains lower training loss from the very beginning of training phase (Fig. 13).

### 5.3 Comparative Study of two-stacked LSTM with Cerebral LSTM

Cerebral LSTM consists of two cell states ( $U_c(t)$  and  $L_c(t)$ ), so we performed another comparative study to see if our proposed cell has an advantage over two-stacked LSTM based recurrent neural networks (Fig. 14).

It can be easily seen that Cerebral LSTM easily outperformed two-stacked LSTM based recurrent neural network. We even conducted further analysis to know whether after 500 epochs two-stacked LSTM outperforms our proposed cell, but it does not happen. The value of training loss of Cerebral LSTM on 500 epochs was 0.3979, which two-stacked LSTM achieved after 678 epochs. This analysis makes it very clear that Cerebral LSTM outperformed two-stacked LSTM.

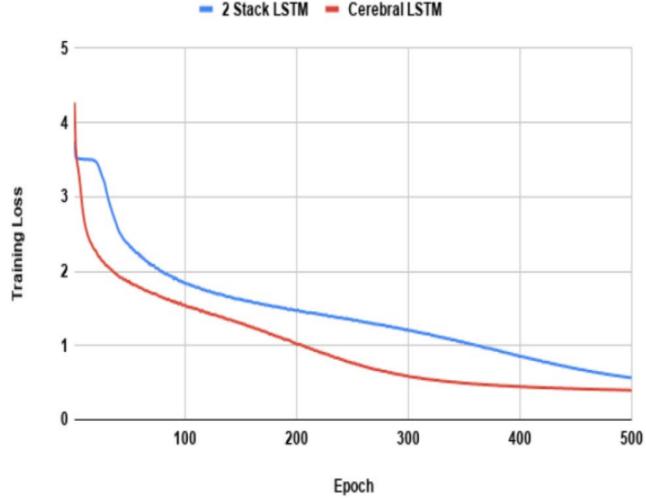


Figure 14: Comparative study of training loss.

## 6 Comparative Study of Generated Data

Character level data is generated and its textual quality is taken into consideration in our comparative study. Data generated by Cerebral LSTM were having better quality than data generated by two-stacked LSTM and single LSTM based recurrent neural networks. This is because after 500 epochs Cerebral LSTM had lower training loss than two-stacked and single LSTMs which made it easier for our proposed cell to better understand the input data during training phase (Table 3).

Single LSTM	Two-stacked LSTM	Cerebral LSTM
Lenny Leonard: I'm sure serve big chief 'am	Buen...	Lenny Leonard: What's the matter Home T?
there	Thief_Wiggum: As a little room, it could pulce would yo, nive?	Moe.Szyslak: (MADDED) Jh...
Lenny Leonard: Ne need givan e	Linder: Ho	

Table 3: Comparative outputs of different LSTM architectures.

## 7 Conclusion

Our proposed recurrent cell ‘Cerebral LSTM’ showed the ability to better understand data and has easily outperformed both single LSTM and two-stacked LSTM based recurrent neural networks. Many variants of Cerebral LSTM can be designed using available varieties of LSTM cells such as peephole LSTM. Further research work can be conducted on designing Cerebral LSTM based stacked recurrent neural networks for designing deep learning architectures for understanding time-series data. Other recurrent cells including gated recurrent units can also be analyzed after modifying its internal connections similar to our cerebral structure.

## References

- [1] R. Kumar, “Cerebral lstm implementation in pytorch.” <https://github.com/mr-ravin/cerebral-lstm>. Last accessed: 16 Jul 2024.
- [2] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [3] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [4] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.
- [5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [6] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [7] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, “An exploration of dropout with lstms,” in *Interspeech*, pp. 1586–1590, 2017.

- [8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [9] A. Huang and R. Wu, “Deep learning for music,” *arXiv preprint arXiv:1606.04930*, 2016.
- [10] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.
- [11] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.
- [12] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, “A comprehensive survey of deep learning for image captioning,” *ACM Computing Surveys (CsUR)*, vol. 51, no. 6, pp. 1–36, 2019.
- [13] R. Kumar, “Cerebral rnn experimental results.” <https://github.com/mr-ravin/cerebral-rnn-experimental-results>. Last accessed: 31 Jan 2019.